Our Docket No.: 51876P367
Express Mail No.: ev339906054us

# UTILITY APPLICATION FOR UNITED STATES PATENT

FOR

# DELAY LOCKED LOOP (DLL) IN SEMICONDUCTOR DEVICE

Inventor(s):
Hea-Suk Jung

BLAKELY, SOKOLOFF, TAYLOR & ZAFMAN LLP
12400 Wilshire Boulevard, Seventh Floor
Los Angeles, California 90025
Telephone: (310) 207-3800

# DELAY LOCKED LOOP (DLL) IN SEMICONDUCTOR DEVICE

## Field of the Invention

The present invention relates to a semiconductor device; and, more particularly, to a delay locked loop having a variable clock divider to be applied to high and low frequencies by using a column address strobe (CAS) latency in the semiconductor device.

## Description of Related Art

Generally, a clock signal is employed to set a timing reference and secure a rapid operation without any error in a system or a circuit. At this time, a time delay is generated when the clock signal provided from the external circuit is used in an internal circuit. The time delay is adjusted by a delay locked loop (DLL) circuit to have an identical phase between the external clock signal and the internal clock signal.

Fig. 1 is a block diagram illustrating a conventional DLL circuit. The DLL circuit includes a clock buffer 110, a clock divider 120, a phase comparator 130, a shift controller 140, a shift register 150, a plurality of delay lines 161, 162 and 163, a delay model unit 170 and a plurality of DLL drivers 181 and 182.

The clock buffer 110 receives a external clock signal and

an inverted clock signal CLK and /CLK and outputs internal clock signals Fall_clk and Rise_clk. The clock divider 120 outputs a divided signal Delay_in having a predetermined pulse width per eight periods of the internal clock signal and a reference signal ref inverting the divided signal Rise_clk by receiving the internal clock signal Rise_clk. The phase comparator 130 compares phases between the reference signal ref and a feedback signal and outputs a comparison signal Pc<0:3>, and the shift controller 140 generates shift control signals SR and SL by receiving the comparison signal Pc<0:3> from the phase comparator 130. The shift register 150 receives the shift control signals SR and SL and selects delay position of the delay lines. The plurality of delay lines 161 to 163, each including a plurality of unit delays, outputs a plurality of delayed signals ifclk and irclk by delaying the internal clock signals Fall_clk and Rise_clk according to outputs of the shift register 150 and a delayed feedback delaying the divided signal Delay_in. The delay model unit 170 generates a feedback signal compensating a time delay of the internal clock signal compared with the external. The plurality of DLL drivers 181 and 182 transmit the plurality of delayed clock signal ifclk and irclk to an internal circuit.

The DLL circuit repeatedly compares a rising edge of the reference signal with that of the feedback signal and, when a jitter between the rising edge of the reference signal and that of the feedback signal is minimized, delay is locked. Therefore, the time difference between the external clock

signal and the internal clock signal is compensated, so that the delayed clock signals ifclk and irclk are synchronized with the external clock signals /CLK and CLK.

Fig. 2 is a timing diagram showing an operation for a low frequency in the conventional DLL circuit.

As shown, the clock divider 120 outputs the divided signal Delay_in having a predetermined pulse width (1tck, herein, 1tck corresponds to one period of the clock signal) per eight periods of the internal clock signal Rise_clk and the reference signal ref inverting the divided signal Rise_clk by receiving the internal clock signal Rise_clk. The feedback signal, which is generated by passing the divided signal Delay_in trough the delay line 163 and the delay model unit 170, is compared with the reference signal ref to thereby reduce a time delay therebetween. The divided clock signal Delay_in is initially delayed as much as a unit delay (for example, about 0.2 ns) to thereby be the delayed feedback signal feedback_dly1. The feedback signal feedback_dly1 is inputted to the delay model for compensating a time delay between the external clock signal and the internal clock signal, so that the feedback signal is generated. The feedback signal Feedback is delayed as much as about 5 ns compared with the delayed feedback signal Feedback_dly1. Namely, the delay model unit 170 has a delay of about 5 ns. Since the divided signal delay_in initially passes the delay line 163 having the delay of about 0.2 ns and the delay model unit 170 having the delay of about 5 ns, a time difference

(Td) between the rising time of the divided signal Delay_in and the rising time of the feedback signal Feedback becomes about 5.2 ns. In a low frequency (for example, tck ≥ 10 ns), the rising edge of the feedback signal Feedback comes before the rising edge of the reference signal ref, which is risen after the predetermined time (1 tck) since the divided signal Delay_in is risen. At this case, the phase comparator 130 generates the comparison signal to increase the number of unit delays, so that a delay of the feedback signal Feedback is adjusted to match an rising edge of the feedback signal Feedback with that of the reference signal ref.

Fig. 3 is a timing diagram showing an operation for a high frequency of the clock divider 120 in the conventional DLL circuit.

As shown, at the high frequency (for example, tck ≤ 5 ns), since the initial divided signal delay_in is passed to the unit delay having the delay of about 0.2 ns and the delay model unit having the delay of about 5 ns, the rising edge of the feedback signal Feedback comes after the rising edge of the reference signal ref. At this case, the phase comparator 130 generates a comparison signal to decrease the number of unit delays, that is, carry out a shift-left operation. However, since the delay lines 16 to 163 cannot be initially shift left, a desired internal clock cannot be obtained.

To solve the above problem, a clock dividing method is introduced as illustrated in Fig. 4.

Referring to Fig. 4, the internal clock signal Rise_clk

is divided to generate a divided signal Delay_in having a predetermined pulse width (2 tck) per eight periods of the internal clock signal and a reference signal ref inverting the divided signal Delay_in. Namely, the rising edge of the reference signal ref comes after 2 tck since the divided signal delay_in is risen. The divided signal delay_in is initially passed to the unit delay having a delay of about 0.2 ns and the delay model unit having a delay of about 5 ns to thereby generate the feedback signal Feedback. Thus, the feedback signal Feedback is delayed as much as about 5.2 ns. Since the rising edge of the reference signal ref comes after 2 tck (= 10 ns) since the divided signal delay_in is risen, the rising edge of the feedback signal Feedback comes before that of the reference signal ref. Thereafter, the phase comparator 130 generates a comparison signal to increase the number of unit delays, that is, carry out a shift-right operation, so that the rising edge of the feedback signal Feedback is matched with the rising edge of the reference signal ref.

Fig. 5 is a circuit diagram showing a conventional 1/8 clock divider. The clocks divider includes a plurality of dividers 510, 520 and 530, and a driver 540.

Fig. 6 is a timing diagram illustrating an operation of the convention 1/8 clock divider in Fig. 5.

Referring to Fig. 6, a first divider 510 receives a clock signal S1 having a predetermine period (tck) and generates a divided signal A toggled with a 2 X tck period by inverting a

phase of the signal S1 when the signal S1 becomes a second logic level (high) and latching a value when the signal S1 becomes a first logic level (low). Thereafter, the second divider 520 receives the divided signal A from the first divider 510 and generates a 1/4 divided signal B having a predetermined pulse width (1 tck) by inverting a phase when the divided signal A becomes a second logic level (high) and maintaining a value when the divided signal A is a first logic level (low). Next, a third divider 530 receives the 1/4 divided signal B and generates a 1/8 divided signal S3 having a predetermined pulse width (1 tck) per eight periods of the clock signal S1. The above mentioned clock divider is suitable only for a low frequency. An inverted 1/8 divided signal S2 is outputted through the driver 540.

Fig. 7 is a circuit diagram illustrating a conventional 2/8 clock divider. The 2/8 clock divider includes a plurality of dividers 710, 720 and 730, and a driver 740.

Fig. 8 is a timing diagram illustrating an operation of the convention 2/8 clock divider in Fig. 5.

Referring to Fig. 8, a first divider 710 receives a clock signal S1 having a predetermine period (tck) and generates a divided signal A toggled with a 2 X tck period by inverting a phase of the signal S1 when the signal S1 becomes a second logic level (high) and latching a value when the signal S1 becomes a first logic level (low). Thereafter, the second divider 720 receives the divided signal A from the first divider 710 and generates a 2/4 divided signal B having a

predetermined pulse width (2 tck) by inverting a phase when the divided signal A becomes a second logic level (high) and maintaining a value when the divided signal A is a first logic level (low). Next, a third divider 730 receives the 2/4 divided signal B and generates a 2/8 divided signal S3 having a predetermined pulse width (2 tck) per eight periods of the clock signal S1. An inverted 2/8 divided signal S2 is outputted through the driver 740. The above mentioned clock divider can be used for a high frequency.

However, if the conventional DLL circuit having the 1/8 clock divider is used, the DLL circuit cannot be used for the high frequency, and, if the convention DLL circuit having the 2/8 clock divider is used, since the reference signal has a pulse width of 2 tck, there is a problem that it takes a long time to achieve a delay locking in a low frequency due to a long rising time difference between the reference signal and the feedback signal Feedback. Also, if the 2/8 clock divider is used in a low frequency, since it causes that the number of unit delays have to be increased so as to achieve a delay locking, there is a problem that a layout area of the DLL circuit is increased.

## Summary of the Invention

It is, therefore, an object of the present invention to provide a delay locked loop (DLL) circuit having a variable clock divider to be applied to high and low frequencies.

In accordance with an aspect of the present invention, there is provided a delay locked loop (DLL) in a semiconductor device, including: an clock buffer receiving an external clock signal and an inverted clock signal and outputting first and second internal clock signals to be used in the DLL circuit; and a variable clock divider receiving the second internal signal from the clock buffer and variably dividing the second internal clock signal to have a predetermined pulse width according to a control signal based on a column address strobe (CAS) latency, which is set according to a frequency of the external clock signal, wherein the control signal is initially set to have a first logic level and is enabled to a second logic level when the CAS latency corresponds to a high frequency.

## Brief Description of the Drawings

The above and other objects and features of the instant invention will become apparent from the following description of preferred embodiments taken in conjunction with the accompanying drawings, in which:

Fig. 1 is a block diagram illustrating a conventional DLL circuit;

Fig. 2 is a timing diagram showing an operation for a low frequency in the conventional DLL circuit;

Fig. 3 is a timing diagram showing an operation that a delay is not locked for a high frequency of the clock divider

in the conventional DLL circuit;

Fig. 4 is a timing diagram showing an operation for a high frequency of the clock divider in the conventional DLL circuit;

Fig. 5 is a circuit diagram showing a conventional 1/8 clock divider;

Fig. 6 is a timing diagram illustrating an operation of the convention 1/8 clock divider in Fig. 5;

Fig. 7 is a circuit diagram illustrating a conventional 2/8 clock divider;

Fig. 8 is a timing diagram illustrating an operation of the convention 2/8 clock divider in Fig. 5;

Fig. 9 is a block diagram illustrating a DLL circuit in accordance with the preferred embodiment of the present invention;

Fig. 10 is a circuit diagram showing the variable clock divider 920 in accordance with the preferred embodiment of the present invention; and

Fig. 11 is a timing diagram illustrating an operation of the delay locked loop in accordance with the preferred embodiment of the present invention.

## Detailed Description of the Invention

Hereinafter, a DLL circuit having a controller for selecting a divider suitable for a random frequency according to the present invention will be described in detail referring

to the accompanying drawings.

Fig. 9 is a block diagram illustrating a DLL circuit in accordance with the preferred embodiment of the present invention. The DLL circuit includes a clock buffer 910, a variable clock divider 920, a phase comparator 930, a shift controller 940, a shift register 950, a plurality of delay lines 961, 962 and 963, a delay model unit 970 and a plurality of DLL drivers 981 and 982.

The clock buffer 910 receives an external clock signal and an inverted clock signal CLK and /CLK and outputs internal clock signals Fall_clk and Rise_clk.

The variable clock divider 920 receives the internal clock signal Rise_clk and a control signal CL based on a column address strobe (CAS) latency, which is set according to a frequency of the external clock signal, wherein the control signal is initially set to have a first logic level (low) and is enabled to a second logic level (high) when the CAS latency corresponds to a high frequency, and outputs a divided signal S2 having a predetermined pulse width and an reference signal S3 inverting the divided signal S2 according to the control signal CL.

The phase comparator 930 receives the feedback signal and the reference signal S3 from the delay model unit 970 and the variable clock divider 920, and compares rising edges therebetween to thereby generate a comparison signal Pc<0:3>. The comparison signal Pc<0:3> is outputted to the shift register controller 940.

The shift controller 940 generates shift control signals SR and SL by receiving the comparison signal Pc<0:3> from the phase comparator 930. The shift register 950 receives the shift control signals SR and SL and selects delay position of the delay lines. If the shift right signal SR of the shift control signals is inputted to the shift register, an activation position of the delay line is moved to right, and if the shift left signal SL of the shift control signals is inputted to the shift register, an activation position of the delay line is moved to left. The plurality of delay lines 961 to 963, each including a plurality of unit delays, outputs a plurality of delayed signals ifclk and irclk by delaying the internal clock signals Fall_clk and Rise_clk according to outputs of the shift register 950 and a delayed feedback signal Feedback_dly1 by delaying the divided signal Delay_in outputted from the variable clock divider 920.

The delay model unit 970 generates a feedback signal compensating a time delay of the internal clock signal compared with the external. The plurality of DLL drivers 981 and 982 transmit the plurality of delayed clock signal ifclk and irclk to an internal circuit.

Fig. 10 is a circuit diagram showing the variable clock divider 920 in accordance with the preferred embodiment of the present invention.

A first divider 1010 receives the second internal clock signal Rise_clk form the clock buffer 910 and generates a first divided signal A by inverting a signal logic level when

the second internal clock signal Rise_clk is a second logic level (high) and maintaining a latched value when the second internal clock signal Rise_clk is a first logic level (low). The first divided signal A is outputted to a second divider 1020.

An operation of the first divider 1010 will be described in detail.

A 1st NAND gate 1011 in the first divider 1010 performs a NAND operation by receiving the second internal clock signal Rise_clk and a 2nd NAND gate 1012 in the first divider 1010 performs a NAND operation by receiving the second internal clock signal Rise_clk. A 1st inverter 1013 inverts the second internal clock signal Rise_clk and a 3rd NAND gate 1014 performs a NAND operation by receiving an output signal of the 2nd NAND gate 1012.

A 4th NAND gate 1015, which is cross-coupled with the 3rd NAND gate 1014, outputs the first divided signal A by performing a NAND operation for an output signal of the 1st NAND gate 1011, and a 5th NAND gate 1016 performs a NAND operation by receiving output signals of the 3rd NAND gate 1014 and the 1st inverter 1013. A 6th NAND gate 1017 performs a NAND operation by receiving output signals of the 4th NAND gate 1015 and the 1st inverter 1013, and a 7th NAND gate 1018 performing an NAND operation by receiving an output signal of the 6th NAND gate 1017 and outputting an output signal to the 2nd NAND gate 1012. A 8th NAND gate 1019, which is cross-coupled with the 7th NAND gate 1018, performs a NAND operation

by receiving an output signal of the 5th NAND gate 1016 and outputs an output signal to the 1st NAND gate 1011.

A second divider 1020 receives the first divided signal A form the first divider 1010 and generates a second dividing signal B-1 by inverting a signal logic level when the first divided signal A is a second logic level (high) and maintaining a latched value when the first divided signal A is a first logic level (low). The second divided signal A is outputted to a selector 1030.

An operation of the second divider 1020 will be described in detail.

In the second divider 1020, a 9th NAND gate 1021 performs a NAND operation by receiving the first divided signal, and a 10th NAND gate 1022 for performing a NAND operation by receiving the first divided signal. A 2nd inverter 1023 inverts the first divided signal, and a 11th NAND gate 1024 performs a NAND operation by receiving an output signal of the 10th NAND gate 1022. A 12th NAND gate 1025, which is cross-coupled with the 11th NAND gate 1024, outputs the second divided signal B-1 by performing a NAND operation for an output signal of the 9th NAND gate 1021.

A 13th NAND gate 1026 performs a NAND operation by receiving output signals of the 11th NAND gate 1024 and the 2nd inverter 1023, and a 14th NAND gate 1027 performs a NAND operation by receiving output signals of the 12th NAND gate 1025 and the 2nd inverter 1023. A 15th NAND gate 1028 performs an NAND operation by receiving an output signal of the 14th

NAND gate 1027 and outputs its output signal to the 10th NAND gate 1022, and a 16th NAND gate 1029, which is cross-coupled with the 15th NAND gate 1028, performs a NAND operation by receiving an output signal of the 13th NAND gate 1026 and outputs its output signal to the 9th NAND gate 1020.

The selector 1030 includes a first pass gate 1031 for passing the second divided signal B-1 to a third divider 1040 when the control signal CL is the first logic level (low), and for breaking the second divided signal when the control signal is the second logic level (high), and a second pass gate for passing a third divided signal B-2 to the third divider when the control signal is the second logic level (high), and for breaking the second divided signal when the control signal is the first logic level (low).

A third divider 1040 receives the second divided signal B-1 or the third divided signal B-2 from the selector 1030 and generates a reference signal S3 by inverting a signal logic level when the second divided signal B-1 or the third divided signal B-2 is a second logic level (high) and maintaining a latched value when the second divided signal B-1 or the third divided signal B-2 is a first logic level (low). The second divided signal A is outputted to the phase comparator 930.

An operation of the third divider 1040 will be described in detail.

In the third divider 1040, a 17th NAND gate 1041 performs a NAND operation by receiving an output signal of the selector 1030, and a 18th NAND gate 1042 performs a NAND operation by

receiving the output signal of the selector. A 3rd inverter 1043 inverts the output signal of the selector 1030, and a 19th NAND gate 1044 performs a NAND operation by receiving an output signal of the 18th NAND gate 1042.

A 20th NAND gate 1045, which is cross-coupled with the 19th NAND gate 1044, performs a NAND operation for an output signal of the 17th NAND gate 1041, and a 21st NAND gate 1046 performs a NAND operation by receiving output signals of the 19th NAND gate 1044 and the 3rd inverter 1043. A 22nd NAND gate 1047 performs a NAND operation by receiving output signals of the 20th NAND gate 1045 and the 3rd inverter 1043 and outputs a reference signal S3. A 23rd NAND gate 1048 performs an NAND operation by receiving an output signal of the 22nd NAND gate 1047 and outputs an output signal to the 18th NAND gate 1042, and a 24th NAND gate 1049, which is cross-coupled with the 23rd NAND gate 1048, performing a NAND operation by receiving an output signal of the 21st NAND gate 1046 and outputs its output signal to the 17th NAND gate 1041.

The driver 1050 outputs a inverted reference signal S2 to the plurality of delay lines 961 to 963 by receiving and inverting the reference signal S3 from the third divider 1040.

Fig. 11 is a timing diagram illustrating an operation of the delay locked loop in accordance with the preferred embodiment of the present invention.

As shown, the first divider 1010 outputs the first divided signal A by receiving the second clock signal Rise_clk and the second divider 1020 outputs the second divided signal

B-1 and the third divided signal B-2 by receiving the first divided signal A. The selector 1030 selectively passes the second divided signal B-1 and the third divided signal B-2 to the third divider 1040. The third divider 1040 outputs the reference signal S3 and the inverted reference signal S2.

Since the clock divider suitable for a corresponding frequency is selected by using a column address strobe (CAS) latency in the delay locked loop (DLL) circuit, the DLL can be used to high and low frequencies. Also, a locking time can be reduced.

While the present invention has been described with respect to the particular embodiments, it will be apparent to those skilled in the art that various changes and modifications may be made without departing from the spirit and scope of the invention as defined in the following claims.